ERNEST ORLANDO LAWRENCE
BERKELEY NATIONAL LABORATORY

Ecole d'ingénieurs et d'architectes de Fribourg
Hochschule für Technik und Architektur Freiburg

Virtual Machine Logbook

10/21/2008

Andrea Cavalli

Julien Poffet

Table of Contents

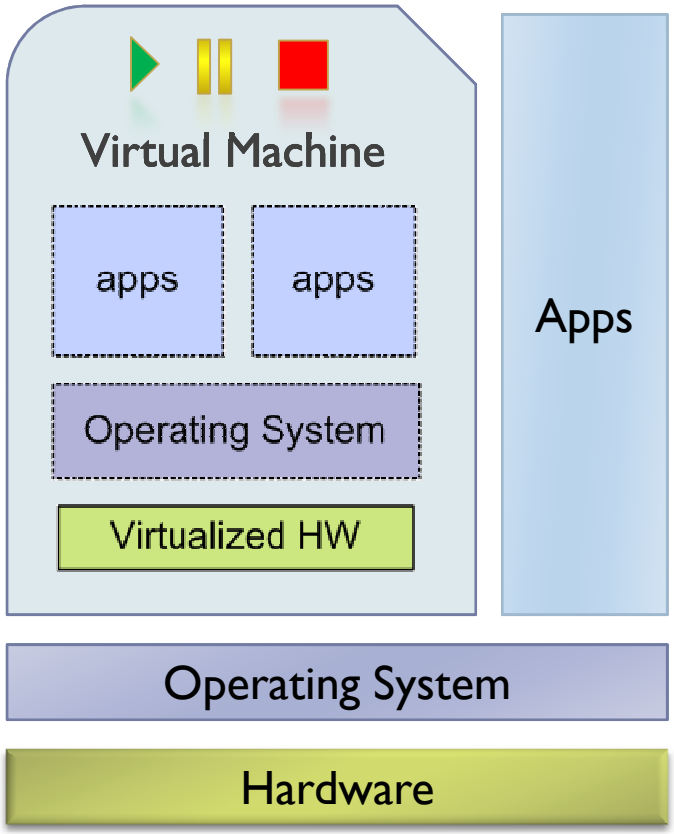
Context – ATLAS

- Software development in the ATLAS experiment
  - Analysis programs are dependent on working environment
    - Environment = ATLAS version, compiler, OS …
    - Different machines = different working environments
    - Same program on two different machines ≠ same behavior
  - Reproduce an environment is hard
  - Sharing the work is difficult

- Idea
  - Use the benefits of virtualization → CernVM

Context – Virtual machine (VM)

- Abstraction of a real computer
  - The environment is contained into the VM
- Operations on virtual machines
  - Start
  - Stop
  - Suspend
  - Snapshot
- Isolation between VM and the host
- Easy to share → consist in a file copy/transfer



Context – CernVM

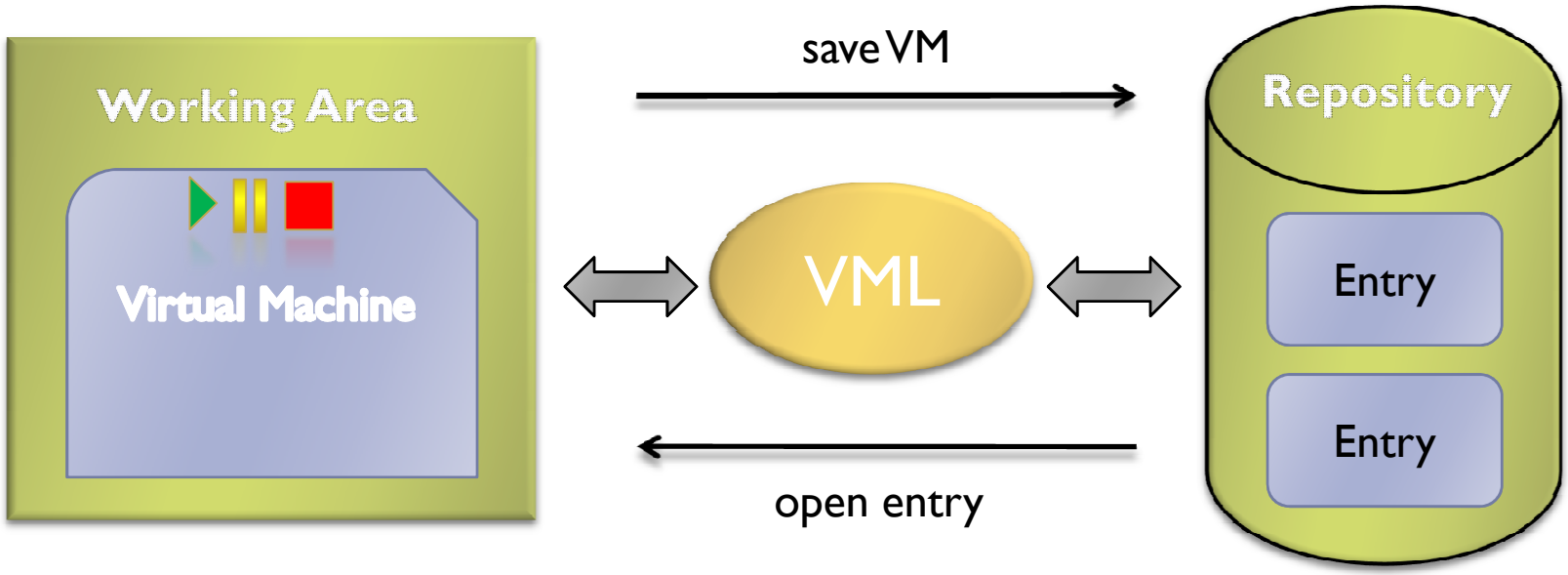
- Idea
  - Develop inside the virtual machine
- Provide virtual machines
  - Working environment (OS + specific software)
  - Used for analysis on LHC data
    - Both for development and production
- CernVM virtual machines are:
  - Configurable
  - Multi-format
    - Independent of hardware and software platform
    - Only a suitable virtual player is required (VMware, Xen…)

VML project goals

- VML = Virtual Machine Logbook
- Provide a simple backup/versioning system
- Keep the size of the backups small
- Increase the usability of CernVM virtual machine
- Simplify the sharing of work environments
- Multi-format

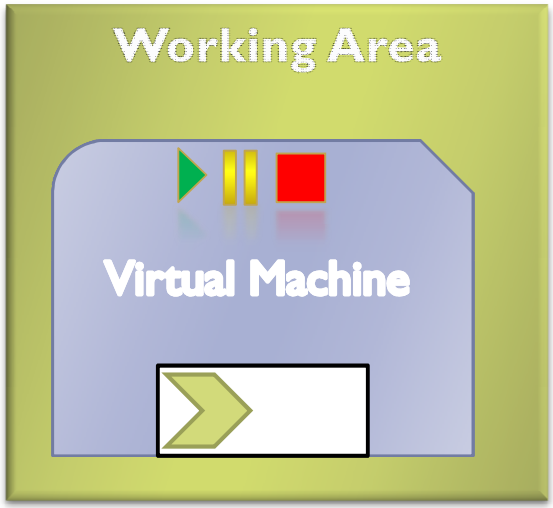
Structure

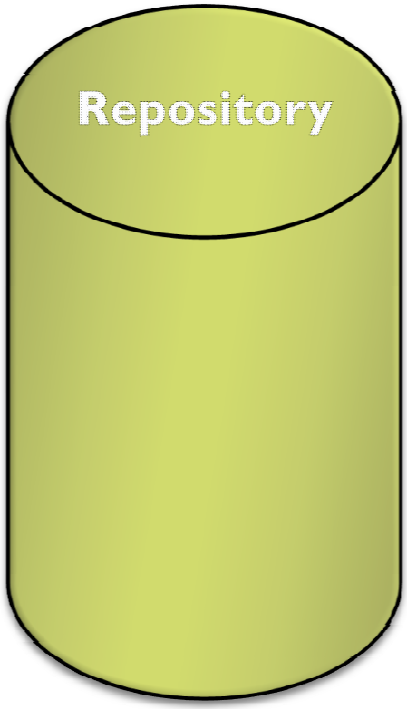
- Working area
  - Provide space to run the virtual machine
- Repository
  - Store the backups of the virtual machines (VML entries)



Save a virtual machine state

```
$> vml.py save entry1
```





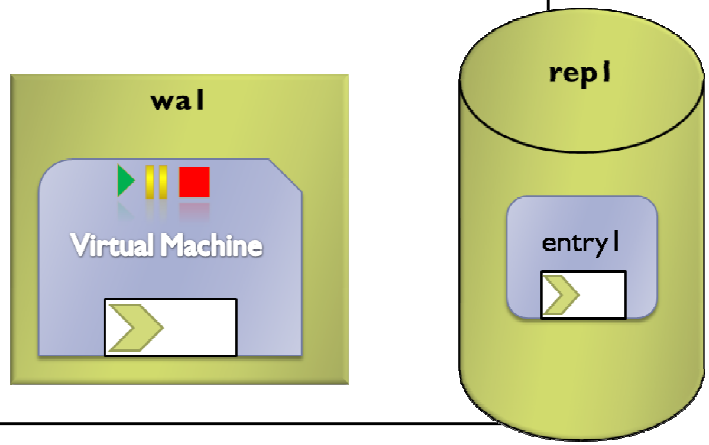
- You can save running virtual machines
  - Save a running VM takes more space in the repository than save a powered off VM

Consult

- Print the content of the repository and of the working areas

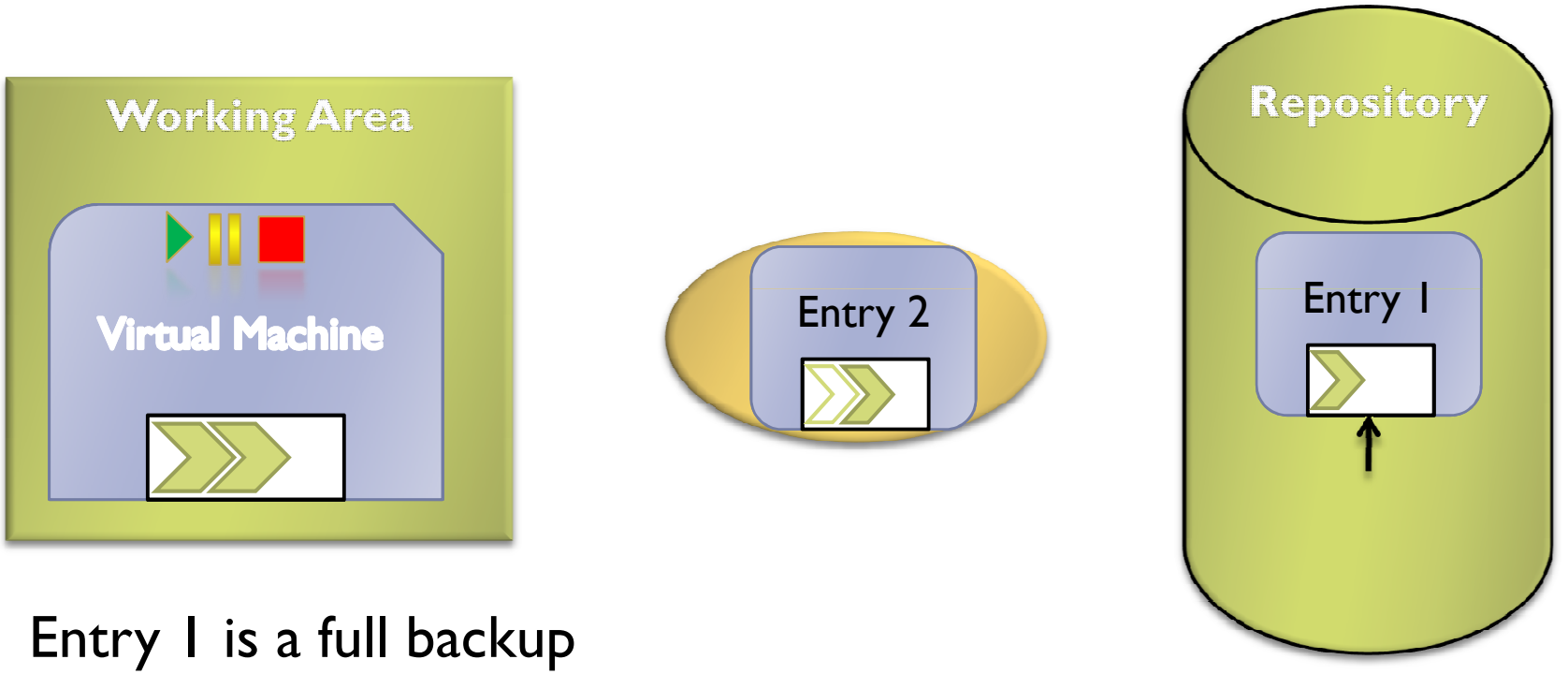
```
$> vml.py consult
```

```
Repositories :
--------------
name            entries          path
----            -------          ----
rep1*           ['entry1']       /scratch-local/poffet/vml/rep1

Working Area:
---------------
name            entry            path
----            -----            ----
wa1             entry1           /scratch-local/poffet/vml/wa1

Entries in repository:
--------------------
+---entry1(10/16/08 - 09:51:25 – full)
```



Save a virtual machine state

```
$> vml.py save entry2
```

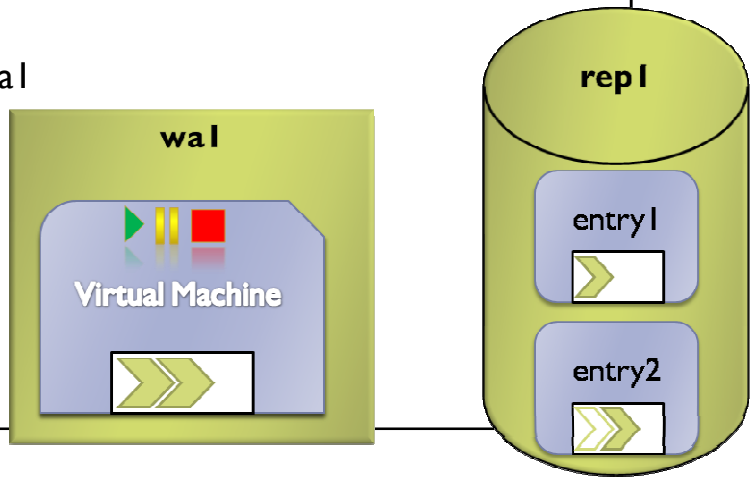


- Entry 1 is a full backup
- Entry 2 is an incremental backup based on Entry 1
  - Contains just the differences from Entry 1 → smaller size

Consult

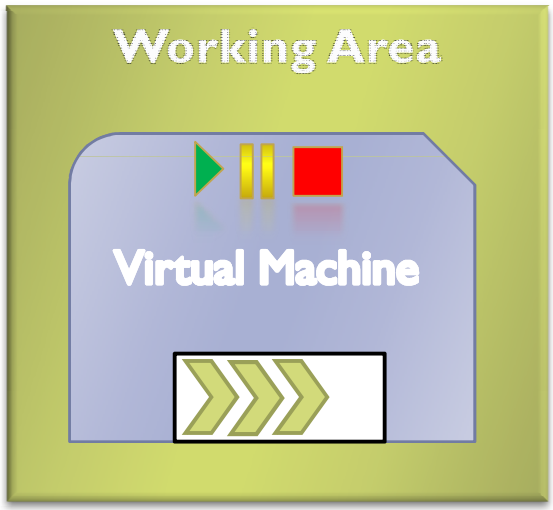
```
$> vml.py consult
```

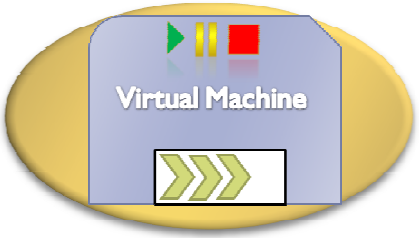
```
Repositories :
--------------
name            entries                 path
----            -------                 ----
rep1*           ['entry1', 'entry2']    /scratch-local/poffet/vml/rep1

Working Area:
---------------
name            entry               path
----            -----               ----
wa1             entry2              /scratch-local/poffet/vml/wa1

Entries in repository:
--------------------
+---entry1(10/16/08 - 09:51:25 - full)
  |
  +---entry2(10/16/08 - 09:58:25 - incremental)
```

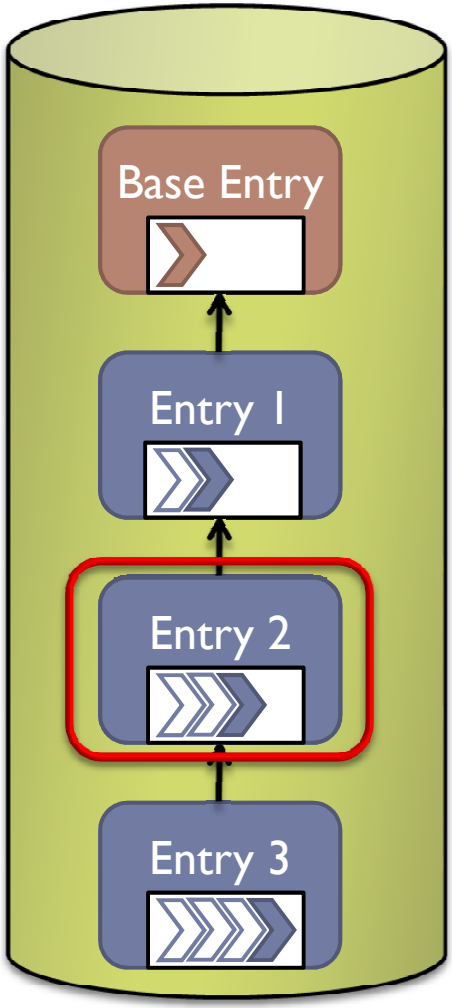


Open an entry in a working area

```
$> vml.py open entry2
```



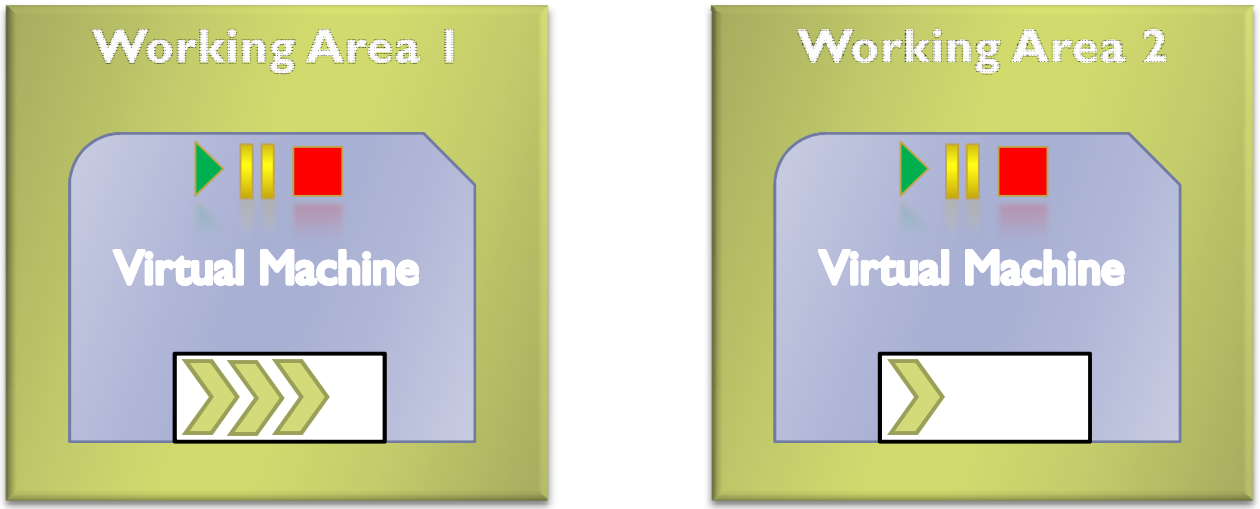




- VML builds the VM and starts it

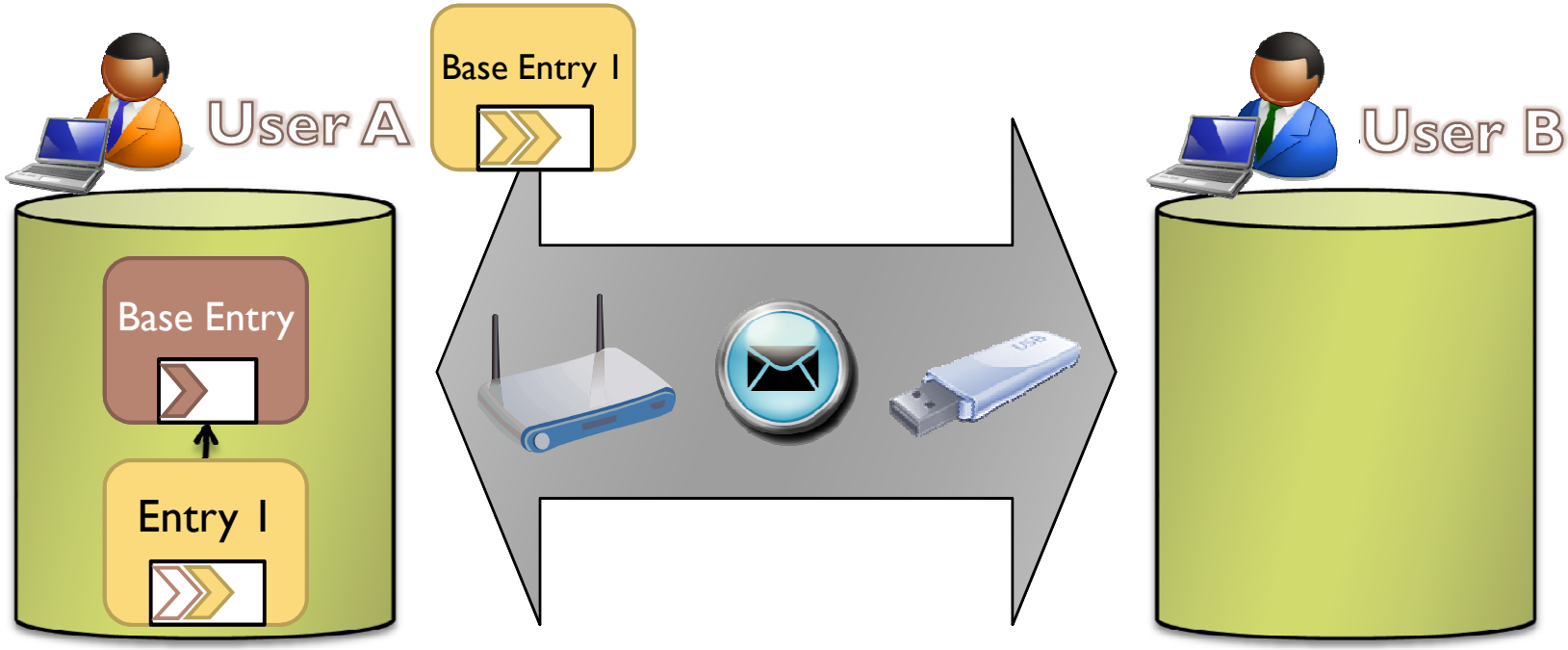
Multiple working areas

- Allow to start multiple VMs at the same time



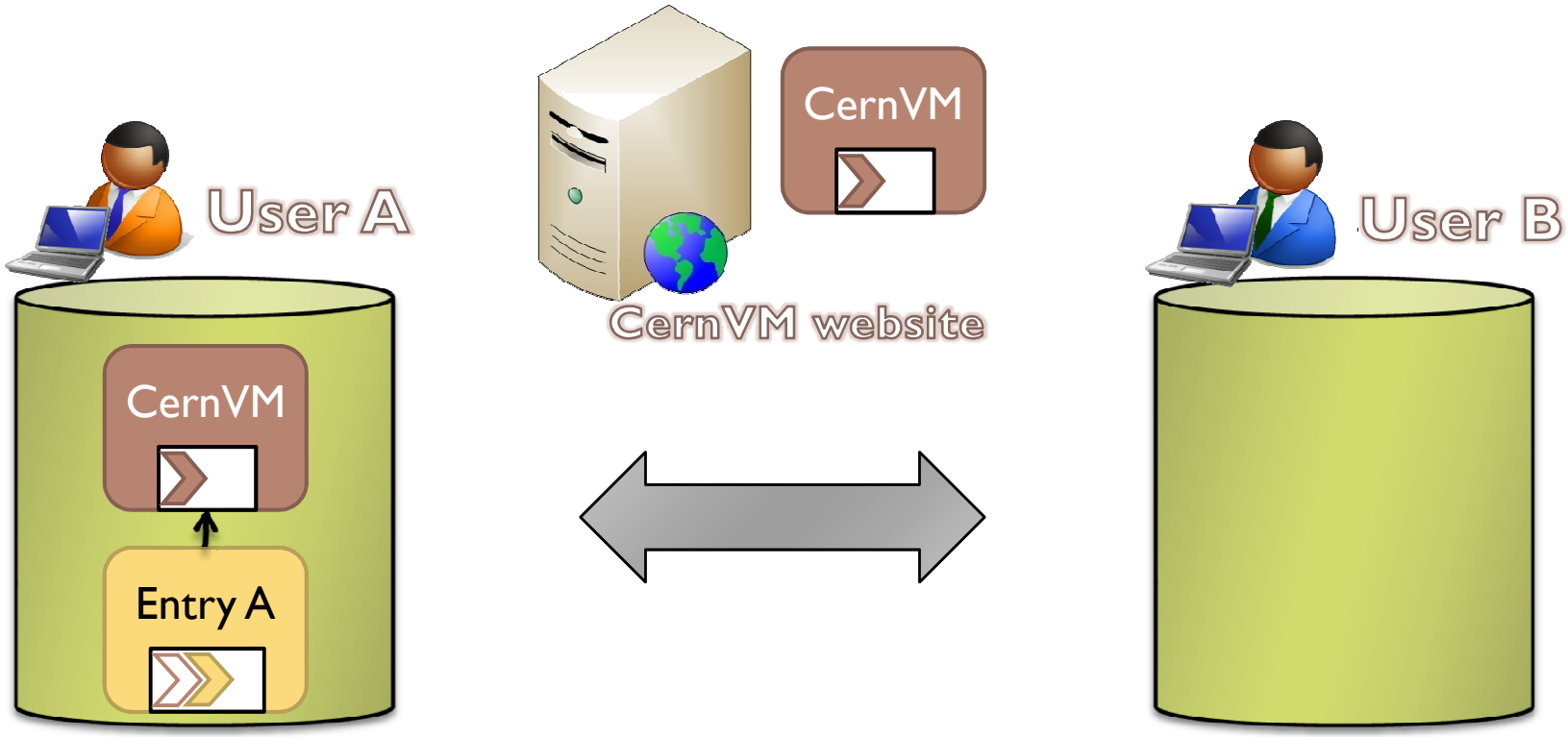
VML now: entries exchange

- Share your entries with other VML users
  - User A exports the entry
  - User B imports the entry: he can now open it (= start the VM)



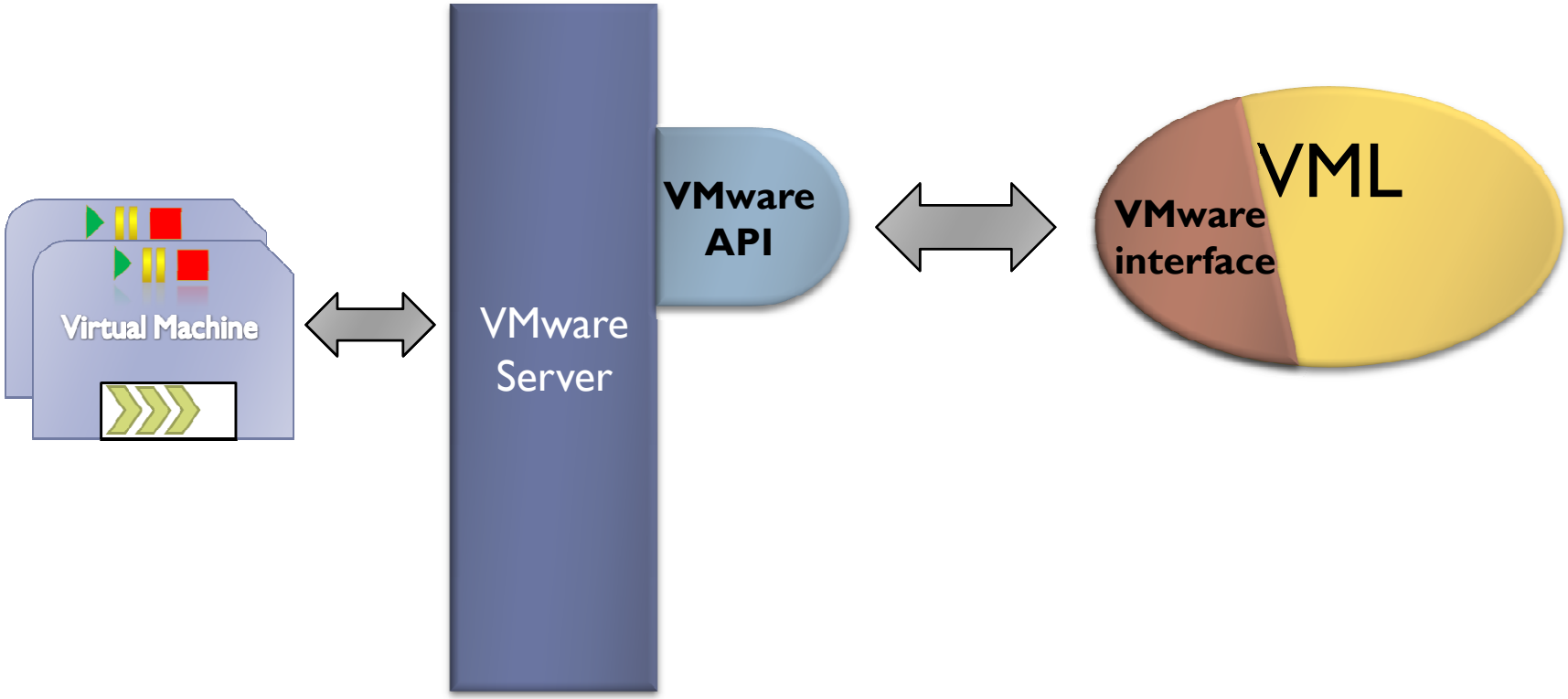
Further work: CernVM integration

- Incremental entries have a common base: CernVM
  - The size of the transferred entries is small (~50 MB or less)
- Automatic CernVM virtual appliances download



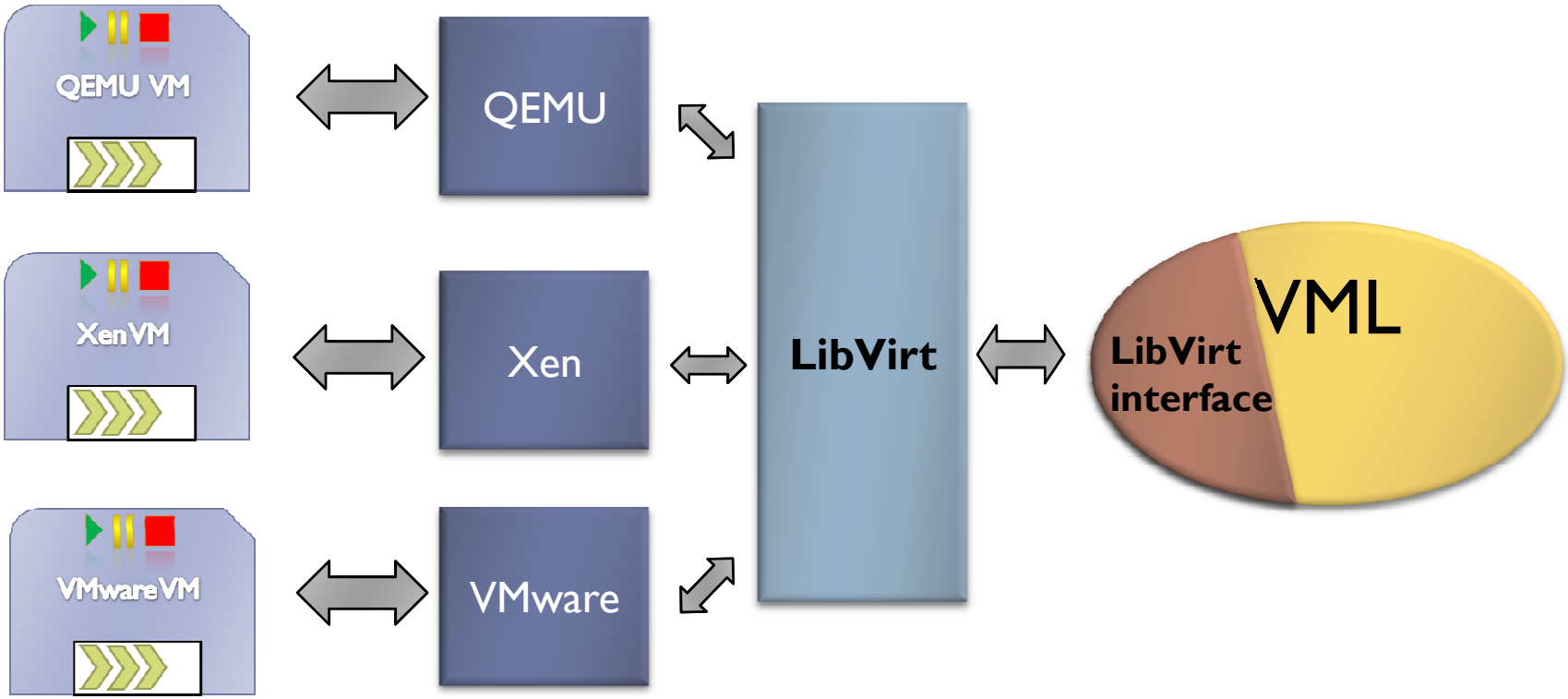
VML now: VMs management

- Done through the VMware Server API
- VML works only with VMware virtual machines



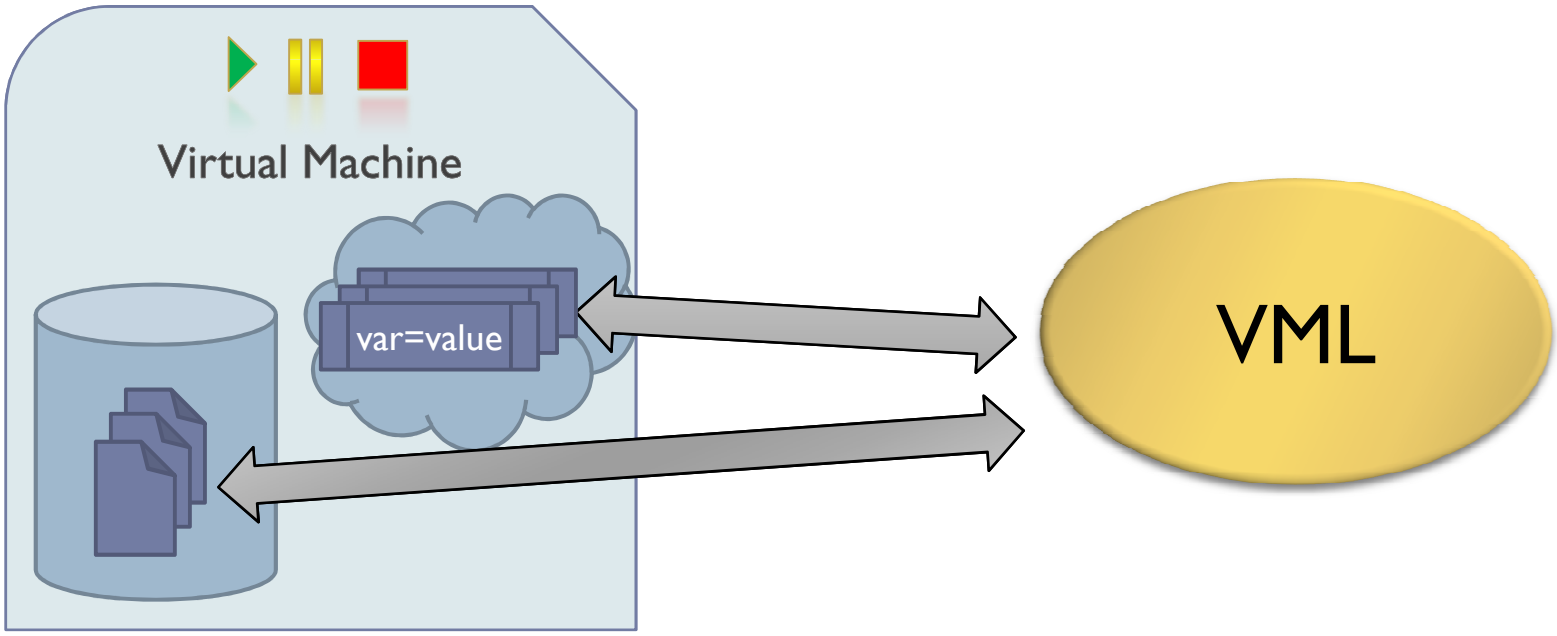
Further work: platform independence

- Support other virtual machines platforms (Xen, QEMU, KVM…)



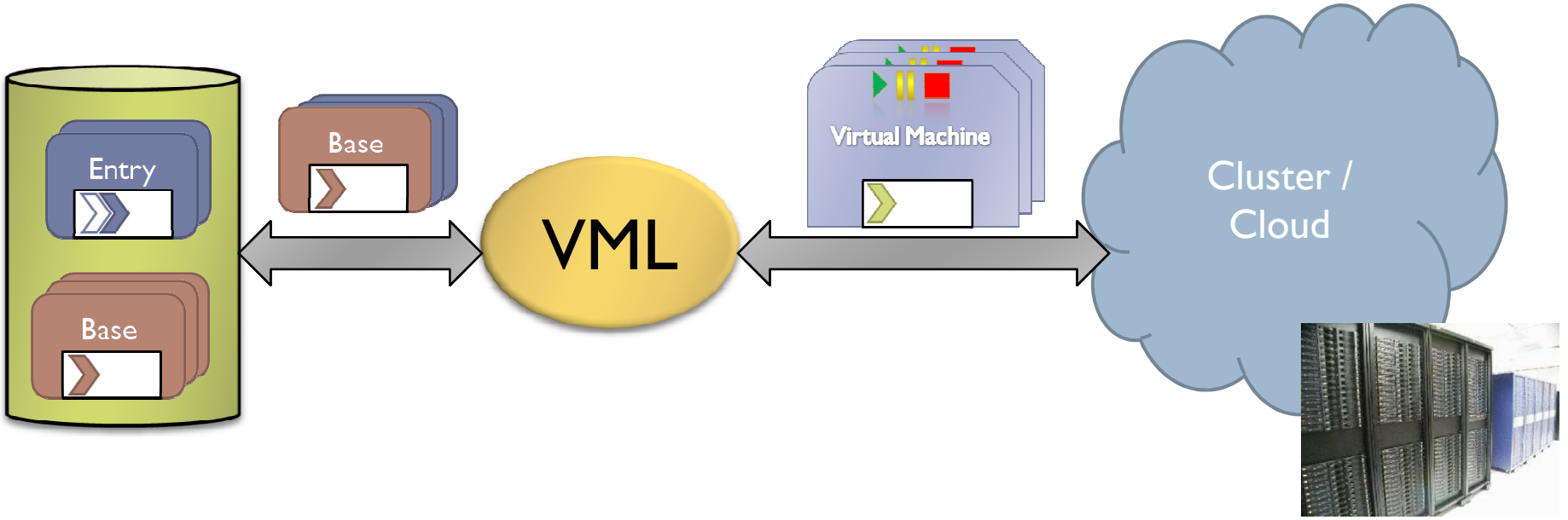
Further work: environment backups

- Backup
  - Environment and relevant files copy
- Restore
  - Push the files and the environment into the virtual machine



Further work

- Use VML to deploy VMs on a cloud or a cluster (like PDSF)



Summary

- Motivation
  - Avoid the issues raised by working on different environments
  - Need to share the work (the virtual machines)
- VML provides the following benefits
  - Simplify the management of the virtual machines
  - Keep a history of the work
  - Easy way to share the work
- Schedule
  - This project will be delivered to ATLAS by February
  - The goal is to have it distributed as a tool with CernVM

Thank you for your attention

- Q & A